

## The computation of the inverse of block-wise centrosymmetric matrices

By CHENGBO LU (Lishui) and CHUANQING GU (Shanghai)

**Abstract.** This paper presents a method to compute the inverse of block-wise centrosymmetric matrices. The method is based on the block diagonalization of each centrosymmetric matrix by a matrix reduced-order method and the application of a recursive algorithm for the inverse of the block diagonal matrices. It is shown that the method is more efficient than the LU decomposition method.

### 1. Introduction

Centrosymmetric matrices arise in the study of certain types of Markov processes because they turn out to be the transition matrices for the processes (see [1], [2] for related references). Such matrices play also an important role in a number of areas such as pattern recognition, antenna theory, mechanical and electrical systems, and quantum physics [3]. As noticed in [4], centrosymmetric matrices appear frequently in the construction of orthonormal wavelet basis. Centrosymmetric matrices arise also in spectral methods in boundary value problems (BVPs)

In recent years the properties and applications of centrosymmetric matrices and their generalizations have been extensively investigated ([5]–[15]). In

---

*Mathematics Subject Classification:* 15A18, 65F05, 65F10.

*Key words and phrases:* block-wise centrosymmetric matrices, reduced form, recursive inverse, LU method, computational complexity.

The first author is the corresponding author, supported by Natural Science Foundation of China (11171137) and Zhejiang Provincial Natural Science Foundation of China (Y6110676).

The second author is supported by Shanghai Natural Science Foundation (10ZR1410900), Key Disciplines of Shanghai Municipality (S30104).

[16]–[18], the recursive algorithms have been applied for the inverse of circulant-structured matrices. Motivated by their works, we focus on the development and investigation of a method for the inverse of block-wise centrosymmetric matrices.

This paper is organized as follows. In Section 2, we review some basic notations and results for centrosymmetric matrices and their generalizations. In Section 3, we develop a method and work out a representative numerical example. In Section 4, we determine the cost of our algorithm and compare it with the LU method. Finally, in Section 5, we present several numerical experiments, exhibiting the accuracy efficiency of the proposed method in terms of CPU time.

## 2. Preliminaries

In this section we begin with some basic notation which frequently used in the sequel.

A matrix  $(a_{i,j})_{p \times q}$  is called a centrosymmetric matrix, if the elements of  $A$  satisfy the relation  $a_{i,j} = a_{p-i+1, q-j+1}$  for all  $1 \leq i \leq p$  and  $1 \leq j \leq q$ .

Let  $J_n = (e_n, e_{n-1}, \dots, e_1)$ , where  $e_i$  denotes the unit vector with  $i$ th entry 1. According to the definition of the centrosymmetric matrix, a matrix  $A \in \mathbb{R}^{n \times n}$  being centrosymmetric is equivalent to the fact that  $J_n A J_n = A$ .

Using an appropriate partition, the centrosymmetric matrices can be expressed as follows:

(1) For the case  $n = 2s$ , a centrosymmetric matrix can be written as the following form

$$A = \begin{bmatrix} B & J_s C J_s \\ C & J_s B J_s \end{bmatrix} \quad (2.1)$$

where each of the block matrices  $B$  and  $C$  is an  $s \times s$  matrix.

(2) For the case  $n = 2s + 1$ , a centrosymmetric matrix can be partitioned into the following form

$$A = \begin{bmatrix} B & J_s b & J_s C J_s \\ a^T & \alpha & a^T J_s \\ C & b & J_s B J_s \end{bmatrix} \quad (2.2)$$

with  $B, C \in \mathbb{R}^{s \times s}, a, b \in \mathbb{R}^{s \times 1}$  and  $\alpha$  a scalar.

If  $n$  is an even number, say  $n = 2s$ , then, by choosing the orthogonal matrix

$$Q = \frac{\sqrt{2}}{2} \begin{bmatrix} I_s & I_s \\ -J_s & J_s \end{bmatrix} \quad (2.3)$$

We can easily obtain the following relation

$$Q^T A Q = \begin{bmatrix} B - J_s C & \\ & B + J_s C \end{bmatrix} \quad (2.4)$$

and

$$A^{-1} = Q \begin{bmatrix} (B - J_s C)^{-1} & \\ & (B + J_s C)^{-1} \end{bmatrix} Q^T,$$

set  $H = (B - J_s C)^{-1}$ ,  $D = (B + J_s C)^{-1}$ , then

$$A^{-1} = \frac{1}{2} \begin{bmatrix} D + H & (D - H)J_s \\ J_s(D - H) & J_s(D + H)J_s \end{bmatrix} \quad (2.5)$$

If  $n$  is odd, i.e.  $n = 2s + 1$ , selecting the orthogonal matrix

$$Q = \frac{\sqrt{2}}{2} \begin{bmatrix} I_s & & I_s \\ & \sqrt{2} & \\ -J_s & & J_s \end{bmatrix} \quad (2.6)$$

then the following equality

$$Q^T A Q = \begin{bmatrix} B - J_s C & & \\ & \alpha & \sqrt{2}a^T \\ & \sqrt{2}J_s b & B + J_s C \end{bmatrix} \quad (2.7)$$

and

$$A^{-1} = Q \begin{bmatrix} (B - J_s C)^{-1} & & \\ & \left( \begin{matrix} \alpha & \sqrt{2}a^T \\ \sqrt{2}J_s b & B + J_s C \end{matrix} \right)^{-1} & \\ & & \end{bmatrix} Q^T$$

set  $H = (B - J_s C)^{-1}$ ,  $D = \left( \begin{matrix} \alpha & \sqrt{2}a^T \\ \sqrt{2}J_s b & B + J_s C \end{matrix} \right)^{-1}$

then

$$A^{-1} = \frac{1}{2} \begin{bmatrix} D(2:s+1,2:s+1)+H & \sqrt{2}D(2:s+1,1) & (D(2:s+1,2:s+1)-H)J_s \\ \sqrt{2}D(1,2:s+1) & 2D(1,1) & \sqrt{2}D(1,2:s+1)J_s \\ J_s(D(2:s+1,2:s+1)-H) & \sqrt{2}J_s D(2:s+1,1) & J_s(D(2:s+1,2:s+1)+H)J_s \end{bmatrix} \quad (2.8)$$

holds.

A block matrix

$$A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mm} \end{pmatrix} = [A_{i,j}]$$

with  $A_{ij}$  centrosymmetric square matrices of order  $n$ , is called a block-wise centrosymmetric matrix ([8]).

Having the above preparation, we can show the following important result immediately.

**Theorem 2.1.** *Let  $A = [A_{i,j}]$  be an  $m \times m$  block-wise centrosymmetric matrix of order  $n$ . Then there holds*

$$A = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix} \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \dots & \Lambda_{1m} \\ \Lambda_{21} & \Lambda_{22} & \dots & \Lambda_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{m1} & \Lambda_{m2} & \dots & \Lambda_{mm} \end{bmatrix} \begin{bmatrix} Q^T & 0 & \dots & 0 \\ 0 & Q^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q^T \end{bmatrix} \quad (2.9)$$

if  $n = 2m$ ,  $Q, \Lambda_{i,j}$  are of the form (2.3), (2.4), respectively, else if  $n = 2m + 1$ , they are of the form (2.6), (2.7), respectively.

Let  $\Lambda = [\Lambda_{i,j}]$ , if  $A$  is invertible then there holds

$$A^{-1} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix} \begin{bmatrix} L_{11} & L_{12} & \dots & L_{1m} \\ L_{21} & L_{22} & \dots & L_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ L_{m1} & L_{m2} & \dots & L_{mm} \end{bmatrix} \begin{bmatrix} Q^T & 0 & \dots & 0 \\ 0 & Q^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q^T \end{bmatrix} \quad (2.10)$$

where  $\Lambda^{-1} = [L_{ij}]$ .

PROOF. A direct application of (2.4) or (2.7) immediately implies the desired result.  $\square$

According to (2.10) the inverse of  $A$  is actually reduced to matrices  $\Lambda = [\Lambda_{ij}]$ . The computation of  $\Lambda^{-1}$  is obtained in Section 3 by applying a recursive algorithm.

### 3. Recursive algorithm

The basic idea of the recursive algorithm lies in the fact that in each step the involved matrices are split into four matrices of the same order. Thus, the algorithm is applied to matrices with  $2^k \times 2^k$  block diagonal matrices. Since the number  $m$  of blocks of  $\Lambda$  is arbitrary, we may consider the matrix  $\Lambda$  as a block of an appropriate block matrix with  $2^k \times 2^k$  block diagonal matrices.

Let  $m$  be the order of blocks of  $\Lambda$  and  $k$  the minimum integer with  $m \leq 2^k$ .

We introduce the (augmented)  $2^k \times 2^k$  block matrix

$$\Delta = \begin{bmatrix} cc\Lambda & 0 \\ 0^T & I \end{bmatrix} \tag{3.1}$$

(with  $\Delta = \Lambda$  for  $m = 2^k$ ) where  $0$  is the  $(mn) \times ((2^k - m)n)$  zero matrix and  $I$  is the identity matrix of order  $(2^k - m)n$ . The inverse  $\Lambda^{-1}$  of  $\Lambda$  will be derived by applying the recursive algorithm analyzed below in matrix  $\Delta$ .

The following lemma summarize some simple basic invertibility properties of  $2 \times 2$  block matrices.

**Lemma 3.1** (see [16]). *Let  $X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$  be a  $2 \times 2$  block matrix with square blocks of the same order and  $A$  invertible. Then the following assertions hold:*

1.  *$X$  is invertible if and only if the Schur complement  $S_X(A) = D - CA^{-1}B$  of  $A$  in  $X$  is invertible. (particular case of Schur complement Lemma)*
2. *If  $X$  is invertible (or equivalently  $S_X(A)$  is invertible), then the inverse  $X^{-1}$  of  $X$  is given by the identity*

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \tag{3.2}$$

In the following, we develop a recursive algorithm, consisting of a forward and backward recurrence procedure, for the inverse of  $\Lambda$ .

**Recursive algorithm**

Forward recurrence procedure

*Step 1:* We split  $\Delta$  into four  $2^{k-1} \times 2^{k-1}$  block matrices

$$\Delta = \begin{bmatrix} X_1^1 & Y_1^1 \\ Z_1^1 & W_1^1 \end{bmatrix},$$

where

$$X_1^1 = \begin{bmatrix} \Delta_{11} & \dots & \Delta_{1(2^{k-1})} \\ \vdots & \ddots & \vdots \\ \Delta_{(2^{k-1})1} & \dots & \Delta_{(2^{k-1})(2^{k-1})} \end{bmatrix},$$

$$Y_1^1 = \begin{bmatrix} \Delta_{1(2^{k-1}+1)} & \dots & \Delta_{1(2^k)} \\ \vdots & \ddots & \vdots \\ \Delta_{(2^{k-1})(2^{k-1}+1)} & \dots & \Delta_{(2^{k-1})(2^k)} \end{bmatrix},$$

$$Z_1^1 = \begin{bmatrix} \Delta_{(2^{k-1}+1)1} & \cdots & \Delta_{(2^{k-1}+1)(2^{k-1})} \\ \vdots & \ddots & \vdots \\ \Delta_{(2^k)1} & \cdots & \Delta_{(2^k)(2^{k-1})} \end{bmatrix},$$

$$W_1^1 = \begin{bmatrix} \Delta_{(2^{k-1}+1)(2^{k-1}+1)} & \cdots & \Delta_{(2^{k-1}+1)(2^k)} \\ \vdots & \ddots & \vdots \\ \Delta_{(2^k)(2^{k-1}+1)} & \cdots & \Delta_{(2^k)(2^k)} \end{bmatrix}$$

Since  $\Delta$  is invertible by hypothesis, if  $X_1^1$  is invertible, then by Lemma 3.1  $S_\Delta(X_1^1)$  is also invertible and hence by the identity (3.2) we conclude that the inverse of the  $2^k \times 2^k$  block matrix  $\Delta$  is reduced to that of the  $2^{k-1} \times 2^{k-1}$  block matrices

$$M_1^1 = X_1^1, \quad M_1^2 = S_\Delta(X_1^1) = W_1^1 - Z_1^1(X_1^1)^{-1}Y_1^1.$$

*Step 2:* We split each one of the latter block matrices  $M_1^j$  ( $j = 1, 2$ ) into four  $2^{k-2} \times 2^{k-2}$  block matrices

$$M_1^1 = \begin{bmatrix} X_2^1 & Y_2^1 \\ Z_2^1 & W_2^1 \end{bmatrix}, \quad M_1^2 = \begin{bmatrix} X_2^2 & Y_2^2 \\ Z_2^2 & W_2^2 \end{bmatrix}.$$

Since, by Step 1, the matrices  $M_1^1$  and  $M_1^2$  are invertible, supposing  $X_2^1$  and  $X_2^2$  are invertible, by Lemma 3.1  $S_{M_1^1}(X_2^1)$  and  $S_{M_1^2}(X_2^2)$  are also invertible. Hence, the computation of the inverses of the matrices  $M_1^j$  by (3.2) requires the inverses of the following  $2^2$  matrices:

$$M_2^j = X_2^j, \quad M_2^{j+2} = S_{M_1^j}(X_2^j) = W_2^j - Z_2^j(X_2^j)^{-1}Y_2^j \quad (j = 1, 2)$$

of order  $2^{k-2}n$ , i.e. that of the original matrix  $\Delta$  divided by four.

*Step i:* We split each one of the  $2^{k-i+1} \times 2^{k-i+1}$  block matrices  $M_{i-1}^j$  ( $j = 1, \dots, 2^{i-1}$ ) of step  $i-1$  into four  $2^{k-i} \times 2^{k-i}$  block matrices

$$M_{i-1}^j = \begin{bmatrix} X_i^j & Y_i^j \\ Z_i^j & W_i^j \end{bmatrix}.$$

By Step  $i-1$  the matrices  $M_{i-1}^j$  are invertible. Thus, if  $X_i^j$  are invertible, then by Lemma 3.1  $S_{M_{i-1}^j}(X_i^j)$  are also invertible and hence for the computation of the inverses of the matrices  $M_{i-1}^j$  by means of (3.2) we have to invert the following  $2^i$  matrices of order  $2^{k-i}n$ :

$$M_i^j = X_i^j, \quad M_i^{j+2^{i-1}} = S_{M_{i-1}^j}(X_i^j) = W_i^j - Z_i^j(X_i^j)^{-1}Y_i^j.$$

In the notation  $M_i^j$  the subscript  $i$  indicates the number of the step and the superscript  $j$  the serial number of the matrices, that have to be inverted in step  $i$ .

...

*Step k:* We split each one of the  $M_{k-1}^j$  into four matrices

$$M_{k-1}^j = \begin{bmatrix} X_k^j & Y_k^j \\ Z_k^j & W_k^j \end{bmatrix}.$$

Since, by Step  $k - 1$ ,  $M_{k-1}^j$  are invertible, and  $X_k^j, Y_k^j, Z_k^j, W_k^j$  are of the form  $\Lambda_{ij}$  in (2.9). If  $X_k^j$  are invertible, then the inverse of the matrices  $M_{k-1}^j$  by the means of (3.2) require the inverses of the following  $2^k$  matrices of order  $n$ :

$$M_k^j = X_k^j, \quad M_k^{j+2^{k-1}} = S_{M_{k-1}^j}(X_k^j) = W_k^j - Z_k^j(X_k^j)^{-1}Y_k^j.$$

which are diagonal block matrices of the form  $\Lambda_{ij}$  in (2.9).

#### Backward recurrence procedure

*Step k:* We first consider the inverses of the diagonal block matrices  $M_k^j$  ( $j = 1, 2, \dots, 2^k$ ) by applying LU method to each block of  $M_k^j$ , then compute  $(M_{k-1}^j)^{-1}$  by the means of

$$\begin{aligned} & (M_{k-1}^j)^{-1} \\ = & \begin{bmatrix} (M_k^j)^{-1} + (M_k^j)^{-1}Y_k^j(M_k^{j+2^{k-1}})^{-1}Z_k^j(M_k^j)^{-1} & -(M_k^j)^{-1}Y_k^j(M_k^{j+2^{k-1}})^{-1} \\ -(M_k^{j+2^{k-1}})^{-1}Z_k^j(M_k^j)^{-1} & (M_k^{j+2^{k-1}})^{-1} \end{bmatrix}. \end{aligned}$$

...

*Step i:* Since the inverse matrices  $(M_i^j)^{-1}$  have been determined in the preceding Step  $i + 1$ , the matrices  $(M_{i-1}^j)^{-1}$  are computed by applying

$$\begin{aligned} & (M_{i-1}^j)^{-1} \\ = & \begin{bmatrix} (M_i^j)^{-1} + (M_i^j)^{-1}Y_i^j(M_i^{j+2^{i-1}})^{-1}Z_i^j(M_i^j)^{-1} & -(M_i^j)^{-1}Y_i^j(M_i^{j+2^{i-1}})^{-1} \\ -(M_i^{j+2^{i-1}})^{-1}Z_i^j(M_i^j)^{-1} & (M_i^{j+2^{i-1}})^{-1} \end{bmatrix}. \end{aligned}$$

The matrices  $(M_{i-1}^j)^{-1}$  are used in step  $i - 1$  for the determination of  $(M_{i-2}^j)^{-1}$

...

*Step 1:* The inverse matrices  $(M_1^j)$  ( $j = 1, 2$ ) have already been determined in the preceding Step 2. Thus, the inverse  $\Delta^{-1}$  of  $\Delta$  is computed by

$$\Delta^{-1} = \begin{bmatrix} (M_1^1)^{-1} + (M_1^1)^{-1}Y_1^1(M_1^2)^{-1}Z_1^1(M_1^1)^{-1} & -(M_1^1)^{-1}Y_1^1(M_1^2)^{-1} \\ -(M_1^2)^{-1}Z_1^1(M_1^1)^{-1} & (M_1^2)^{-1} \end{bmatrix}.$$

Summarizing, we notice that the basic idea of the recursive algorithm lies in the fact that following the forward recurrence procedure the matrix inverses involved in Step  $i$  are transferred to step  $i + 1$ . Then, in the last Step  $k$  the involved matrix inverses are obtained by computing the inverses of two  $\frac{n}{2} \times \frac{n}{2}$  matrices. Hence, following the backward recurrence procedure the matrix inverses of Step  $i$  are obtained by using the already determined inverses of Step  $i + 1$  and performing the multiplications provided by the respective identity of (3.2).

Now, since the inverse of  $\Delta$  is

$$\Delta^{-1} = \begin{bmatrix} \Lambda^{-1} & 0 \\ 0^T & I \end{bmatrix} \tag{3.3}$$

we have  $\Lambda^{-1} = \Delta^{-1}$  for  $m = 2^k$ , the inverse  $\Lambda^{-1}$  for  $m < 2^k$  is derived as the block  $\Delta^{-1} (1 : mn, 1 : mn)$  of  $\Delta^{-1}$ .

The recursive algorithm has been applied to matrix  $\Lambda$  instead of  $A$ , because this has smaller computational cost (for details see Section 4).

**Numerical example**

The following representative numerical example indicates and clarifies the application of the developed recursive method for the inverse of a block-wise centrosymmetric matrix.

*Example 1.* The inverse of the  $3 \times 3$  block-wise centrosymmetric matrix of order 4.

$$A = \begin{bmatrix} \begin{pmatrix} 2 & \frac{11}{10} & \frac{9}{10} & 1 \\ 2 & \frac{31}{20} & -\frac{29}{20} & 0 \\ 0 & -\frac{29}{20} & \frac{31}{20} & 2 \\ 1 & \frac{9}{10} & \frac{11}{10} & 2 \end{pmatrix} & \begin{pmatrix} \frac{3}{2} & \frac{3}{10} & \frac{1}{10} & -\frac{5}{2} \\ -3 & 2 & 1 & 1 \\ 1 & 1 & 2 & -3 \\ -\frac{5}{2} & \frac{1}{10} & \frac{3}{10} & \frac{3}{2} \end{pmatrix} & \begin{pmatrix} -\frac{19}{10} & 0 & -1 & -\frac{21}{10} \\ -\frac{1}{5} & \frac{7}{2} & \frac{1}{2} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{2} & \frac{7}{2} & -\frac{1}{5} \\ -\frac{21}{10} & -1 & 0 & -\frac{19}{10} \end{pmatrix} \\ \begin{pmatrix} \frac{7}{5} & \frac{3}{4} & \frac{1}{4} & -\frac{3}{5} \\ \frac{14}{5} & \frac{59}{20} & \frac{61}{20} & -\frac{16}{5} \\ -\frac{16}{5} & \frac{61}{20} & \frac{59}{20} & \frac{14}{5} \\ -\frac{3}{5} & \frac{1}{4} & \frac{3}{4} & \frac{7}{5} \end{pmatrix} & \begin{pmatrix} \frac{5}{2} & \frac{8}{5} & -\frac{7}{5} & -\frac{3}{2} \\ 0 & 4 & -1 & -2 \\ -2 & -1 & 4 & 0 \\ -\frac{3}{2} & -\frac{7}{5} & \frac{8}{5} & \frac{5}{2} \end{pmatrix} & \begin{pmatrix} 3 & \frac{11}{10} & \frac{9}{10} & -2 \\ -\frac{3}{10} & -\frac{6}{5} & \frac{4}{5} & -\frac{7}{10} \\ -\frac{7}{10} & \frac{4}{5} & -\frac{6}{5} & -\frac{3}{10} \\ -2 & \frac{9}{10} & \frac{11}{10} & 3 \end{pmatrix} \\ \begin{pmatrix} 0 & \frac{1}{4} & \frac{3}{4} & -3 \\ 2 & -\frac{1}{2} & -\frac{3}{2} & -2 \\ -2 & -\frac{3}{2} & -\frac{1}{2} & 2 \\ -3 & \frac{3}{4} & \frac{1}{4} & 0 \end{pmatrix} & \begin{pmatrix} \frac{3}{2} & -\frac{3}{2} & -\frac{7}{2} & \frac{5}{2} \\ -2 & -\frac{7}{20} & \frac{3}{20} & 1 \\ 1 & \frac{3}{20} & -\frac{7}{20} & -2 \\ \frac{5}{2} & -\frac{7}{2} & -\frac{3}{2} & \frac{3}{2} \end{pmatrix} & \begin{pmatrix} 2 & \frac{1}{10} & -\frac{3}{10} & -1 \\ -\frac{5}{2} & \frac{4}{5} & -\frac{6}{5} & \frac{7}{2} \\ \frac{7}{2} & -\frac{6}{5} & \frac{4}{5} & -\frac{5}{2} \\ -1 & -\frac{3}{10} & \frac{1}{10} & 2 \end{pmatrix} \end{bmatrix}$$



is

$$A^{-1} = \begin{pmatrix} \begin{pmatrix} 827 & 1683 & 473 & 1297 & 755 & 436 & 318 & 491 & 371 & 5984 & 961 & 1715 \\ -686 & 900 & 900 & 918 & 404 & -891 & 679 & -374 & -691 & 4867 & -1104 & 3987 \\ -892 & -1639 & 756 & 2745 & 576 & 113 & 1168 & 2146 & -1158 & 1353 & 1283 & 497 \\ -509 & -2382 & 1163 & 1502 & 257 & 222 & 1745 & 1047 & -1327 & 986 & 1085 & 522 \\ 2745 & 756 & 1639 & 892 & 2146 & 1168 & 113 & 576 & 497 & 1283 & 1353 & 1158 \\ 1502 & -2382 & -509 & -509 & -1047 & 1745 & -222 & 257 & 522 & -1085 & 986 & -1327 \\ 1297 & 473 & -1683 & 827 & -491 & 318 & -436 & 755 & 1715 & 961 & 5984 & -371 \\ 918 & 900 & 2327 & 686 & -374 & 679 & 891 & 404 & 3987 & -1104 & 4867 & 691 \end{pmatrix} & \begin{pmatrix} 320 & -5343 & -321 & -589 & 431 & -616 & -2177 & 20 & 524 & 843 & -743 & 227 \\ -5343 & 960 & 2411 & -7389 & -2272 & 1416 & -28664 & 63357 & -2895 & -2648 & -1799 & -4980 \\ 437 & 2344 & -3172 & 3172 & -4727 & 1627 & -857 & 902 & 627 & 8498 & 641 & 1795 \\ -3467 & -3677 & 2411 & 960 & 2499 & -815 & -857 & -4727 & 815 & -3999 & 438 & -1647 \\ -1426 & -3172 & 2344 & 437 & 902 & -857 & 1627 & -1684 & -1795 & 641 & -8498 & 627 \\ 563 & 539 & 321 & -1988 & 20 & 2177 & 616 & 431 & 1647 & 438 & -3999 & 815 \\ -6563 & -7389 & -1988 & -5343 & 63357 & -28664 & -5637 & 2272 & 4980 & -1799 & 2648 & -2895 \end{pmatrix} & \begin{pmatrix} 1268 & 644 & 1080 & 3307 & 1472 & -262 & 413 & -1843 & 443 & 921 & -229 & 397 \\ 1605 & -1471 & 4627 & 4429 & 1249 & -1049 & 1151 & 2723 & -1540 & 1169 & 780 & 1798 \\ 935 & 1355 & 2072 & -3503 & -158 & 389 & -713 & 1775 & 1137 & 713 &td> 1956 & 3827 & 3827 \\ 518 & 1428 & -3351 & -1938 & -75 & 735 & -1285 & 912 & 1277 & 561 & 1607 & -5376 \\ -3503 & -2072 & 1355 & 935 & 1775 & -713 & 389 & -158 & -3827 & 1956 & 713 & 1137 \\ 1938 & 3351 & 1428 & 518 & 912 & -1285 & 735 & 75 & -5376 & 1607 & 561 & 1277 \\ 3307 & 1080 & -644 & 1268 & -1843 & 413 & -262 & 1472 & 397 & 229 & 921 & 443 \\ 4429 & 4627 & -1471 & 1605 & -2723 & 1151 & -1049 & 1249 & 1798 & -780 & 1169 & -1540 \end{pmatrix} \end{pmatrix}$$

1. The block matrix  $\Lambda = [\Lambda_{ij}]$  in (2.9) is

$$\Lambda = \begin{bmatrix} \begin{pmatrix} 1 & \frac{1}{5} & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 2 & \frac{1}{10} \end{pmatrix} & \begin{pmatrix} 4 & \frac{1}{5} & 0 & 0 \\ -4 & 1 & 0 & 0 \\ 0 & 0 & -1 & \frac{2}{5} \\ 0 & 0 & -2 & 3 \end{pmatrix} & \begin{pmatrix} \frac{1}{5} & 1 & 0 & 0 \\ -\frac{2}{5} & 3 & 0 & 0 \\ 0 & 0 & -4 & -1 \\ 0 & 0 & 0 & 4 \end{pmatrix} \\ \begin{pmatrix} 2 & \frac{1}{2} & 0 & 0 \\ 6 & -\frac{1}{10} & 0 & 0 \\ 0 & 0 & \frac{4}{5} & 1 \\ 0 & 0 & -\frac{2}{5} & 6 \end{pmatrix} & \begin{pmatrix} 4 & 3 & 0 & 0 \\ 2 & 5 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{5} \\ 0 & 0 & -2 & 3 \end{pmatrix} & \begin{pmatrix} 5 & \frac{1}{5} & 0 & 0 \\ \frac{2}{5} & -1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & -1 & -\frac{2}{5} \end{pmatrix} \\ \begin{pmatrix} 3 & -\frac{1}{2} & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 0 & 0 & -3 & 1 \\ 0 & 0 & 0 & -2 \end{pmatrix} & \begin{pmatrix} -1 & 2 & 0 & 0 \\ -3 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 4 & -5 \\ 0 & 0 & -1 & -\frac{1}{5} \end{pmatrix} & \begin{pmatrix} 3 & \frac{2}{5} & 0 & 0 \\ -6 & 2 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{5} \\ 0 & 0 & 1 & -\frac{2}{5} \end{pmatrix} \end{bmatrix}$$

2. The recursive algorithm for the determination of the matrix  $\Lambda^{-1}$  is applied as follows:

Forward recurrence procedure

*Step 1:* We split  $\Delta$  into four  $2^1 \times 2^1$  block matrices

$$\Delta = \begin{bmatrix} X_1^1 & Y_1^1 \\ Z_1^1 & W_1^1 \end{bmatrix},$$

where

$$X_1^1 = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}, \quad Y_1^1 = \begin{bmatrix} \Lambda_{13} & 0 \\ \Lambda_{23} & 0 \end{bmatrix}, \quad Z_1^1 = \begin{bmatrix} \Lambda_{31} & \Lambda_{32} \\ 0 & 0 \end{bmatrix}, \quad W_1^1 = \begin{bmatrix} \Lambda_{33} & 0 \\ 0 & I \end{bmatrix}.$$

where  $0$  is the  $4 \times 4$  zero matrix and  $I$  is the identity matrix of order 4. To compute the inverse of the matrix  $\Delta$  by means of (3.2) requires the inverse of the  $2^1 \times 2^1$  block matrices with diagonal blocks of order 4

$$M_1^1 = X_1^1, \quad M_1^2 = S_\Delta(X_1^1) = W_1^1 - Z_1^1(X_1^1)^{-1}Y_1^1.$$

*Step 2:* We split  $M_1^j$  into four matrices of order 4

$$M_1^j = \begin{bmatrix} X_2^j & Y_2^j \\ Z_2^j & W_2^j \end{bmatrix}.$$

For the inverse of the matrices  $M_1^j$  by means of (3.2) we have to invert the following 4 diagonal blocks matrices with order 4:

$$M_2^j = X_2^j, \quad M_2^{j+2} = S_{M_1^j}(X_2^j) = W_2^j - Z_2^j(X_2^j)^{-1}Y_2^j.$$

Backward recurrence procedure

*Step 2:* We compute the diagonal block matrix

$$M_2^1 = X_2^1 = \begin{bmatrix} 1 & \frac{1}{5} & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 2 & \frac{1}{10} \end{bmatrix},$$

by applying LU method to each block of  $M_2^1$ , we get

$$(M_2^1)^{-1} = \begin{bmatrix} \frac{15}{13} & -\frac{1}{13} & 0 & 0 \\ -\frac{10}{13} & \frac{5}{13} & 0 & 0 \\ 0 & 0 & -\frac{1}{37} & \frac{20}{37} \\ 0 & 0 & \frac{20}{37} & -\frac{30}{37} \end{bmatrix}.$$

Then

$$M_2^3 = W_2^1 - Z_2^1(X_2^1)^{-1}Y_2^1 = \begin{bmatrix} -\frac{46}{13} & \frac{67}{26} & 0 & 0 \\ -28 & \frac{46}{10} & 0 & 0 \\ 0 & 0 & \frac{141}{185} & \frac{1043}{925} \\ 0 & 0 & -\frac{1648}{185} & \frac{2321}{137} \end{bmatrix}.$$

By applying LU method to each block of  $M_2^3$ , we get

$$(M_2^3)^{-1} = \begin{bmatrix} \frac{533}{7494} & -\frac{154}{3445} & 0 & 0 \\ \frac{1820}{3747} & -\frac{230}{3747} & 0 & 0 \\ 0 & 0 & \frac{1673}{2267} & -\frac{89}{1812} \\ 0 & 0 & \frac{279}{719} & \frac{141}{4247} \end{bmatrix}.$$

By applying (3.2), we get

$$\begin{aligned} (M_1^1)^{-1} &= \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} (M_2^1)^{-1} + (M_2^1)^{-1}\Lambda_{12}(M_2^3)^{-1}\Lambda_{21}(M_2^1)^{-1} & -(M_2^1)^{-1}\Lambda_{12}(M_2^3)^{-1} \\ -(M_2^3)^{-1}\Lambda_{21}(M_2^1)^{-1} & (M_2^3)^{-1} \end{bmatrix} \end{aligned}$$

$$= \left[ \begin{array}{c} \left( \begin{array}{cccc} \frac{455}{1249} & \frac{203}{3747} & 0 & 0 \\ \frac{200}{1249} & \frac{350}{1249} & 0 & 0 \\ 0 & 0 & \frac{414}{1753} & \frac{270}{4247} \\ 0 & 0 & \frac{408}{4247} & -\frac{610}{4247} \end{array} \right) & \left( \begin{array}{cccc} -\frac{755}{1777} & \frac{437}{1904} & 0 & 0 \\ \frac{270}{1249} & -\frac{240}{1249} & 0 & 0 \\ 0 & 0 & \frac{353}{2310} & -\frac{447}{4247} \\ 0 & 0 & \frac{264}{4247} & \frac{538}{4247} \end{array} \right) \\ \left( \begin{array}{cccc} \frac{220}{1249} & -\frac{94}{3747} & 0 & 0 \\ -\frac{630}{1249} & -\frac{185}{3747} & 0 & 0 \\ 0 & 0 & -\frac{303}{1358} & \frac{126}{4247} \\ 0 & 0 & -\frac{685}{2214} & \frac{1340}{4247} \end{array} \right) & \left( \begin{array}{cccc} \frac{533}{7494} & -\frac{154}{3445} & 0 & 0 \\ \frac{1820}{3747} & -\frac{230}{3747} & 0 & 0 \\ 0 & 0 & \frac{1673}{2267} & -\frac{89}{1812} \\ 0 & 0 & \frac{279}{719} & \frac{141}{4247} \end{array} \right) \end{array} \right]$$

Using the same method, we get

$$(M_1^2)^{-1} = \left[ \begin{array}{c} \left( \begin{array}{cccc} -\frac{892}{1777} & \frac{810}{749} & 0 & 0 \\ \frac{1003}{626} & -\frac{1152}{463} & 0 & 0 \\ 0 & 0 & -\frac{476}{7885} & \frac{560}{1133} \\ 0 & 0 & \frac{450}{2521} & -\frac{480}{8927} \end{array} \right) & \left( \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right) \\ \left( \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right) & \left( \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) \end{array} \right]$$

*Step 1:* According to Step 1 of the forward recurrence procedure the matrix  $\Delta^{-1}$  is given by

$$\Delta^{-1} = \begin{bmatrix} (M_1^1)^{-1} + (M_1^1)^{-1}Y_1^1(M_1^2)^{-1}Z_1^1(M_1^1)^{-1} & -(M_1^1)^{-1}Y_1^1(M_1^2)^{-1} \\ -(M_1^2)^{-1}Z_1^1(M_1^1)^{-1} & (M_1^2)^{-1} \end{bmatrix}$$



3. By substituting the determined inverse  $\Lambda^{-1}$  in (2.10) we obtain the inverse of  $A$ .

#### 4. Computational complexity of the inverse

The computational complexity, which we consider, expresses the total number of required scalar multiplications. For the determination of the computational complexity of the method, we need the complexity of the block diagonalization of  $A$ , the recursive algorithm for the inverse of  $\Lambda$  and the block diagonalization of  $\Lambda^{-1}$ .

First, we consider the complexity of the block diagonalization of  $A$  coincides with that of the determination of  $\Lambda_{ij}$ . From (2.4) and (2.7), we can see that  $\Lambda_{ij}$  are determined only requiring 2 scalar–matrix multiplications, they need  $O(n^2)$  multiplications. Thus, the complexity of the block diagonalization of  $A$  is  $m^2O(n^2)$ .

Similarly, the complexity of the block diagonalization of  $\Lambda^{-1}$  is  $m^2O(n^2)$ .

Furthermore, we consider the computational complexity of the recursive algorithm, analyzed in Section 3, is determined by a  $k$ -steps process, based on the backward recurrence procedure. For the description of this process first we need the following basic facts.

For simplicity, we restrict the case of even  $n = 2s$  in the following. There are similar results for odd  $n$ .

Let  $X = [X_{i,j}]$ ,  $Y = [Y_{i,j}]$  be two  $p \times p$  block matrices, and  $X_{i,j}$ ,  $Y_{i,j}$  are of the form (2.4) of order  $n$ . Then we get the following facts.

(1) The computational cost of the product  $XY$  is  $\frac{1}{4}p^3n^3$  scalar multiplications.

(2) The basic identity (3.2) requires 6 matrix multiplications of type  $X^{-1}Y$  or  $YX^{-1}$ , where  $X^{-1}$  is of form (2.4). Hence, the computational cost of (3.2) is  $\frac{3}{2}p^3n^3$  multiplications.

**Lemma 4.1.** *Let  $A = [A_{ij}]$  be an invertible  $m \times m$  block-wise centrosymmetric matrices of order  $mn$  and  $\Lambda = [\Lambda_{ij}]$  be the block matrix of the block diagonalization of  $A$ . We consider the minimum integer  $k$  with  $m \leq 2^k$ . Then, the complexity of the recursive algorithm for the inverse of the  $2^k \times 2^k$  block matrix is*

$$C_R(k, n) = O\left(\frac{1}{4}(2^{3k} + 3 \cdot 2^k)n^3\right).$$

PROOF. According to the basic idea of the recursive algorithm, following the forward recurrence procedure, at Step  $i$  ( $i = 1, 2, \dots, k$ ) we have to invert  $2^i$

matrices by means of (3.2). At the last Step  $k$ , we have to invert  $2^k$  diagonal block matrices  $M_k^j$  ( $j = 1, 2, \dots, 2^k$ ). In the sequel after the analytical inverses of the latter  $2^k$  block diagonal matrices, we follow the backward recurrence procedure. Thus, the matrix inverses of Step  $i$  are obtained by using the already determined inverses of the preceding Step  $i + 1$  and performing (according to (2))  $6 \cdot 2^{i-1}$  matrix multiplications of type  $X^{-1}Y$  or  $YX^{-1}$  with  $X$  and  $Y$  matrices of order  $2^{k-i}n$ . The results of the above discussion are summarized in Table 1.

The determination of the algorithm's computational complexity starts from the last Step  $k$  following the backward recurrence procedure. At Step  $k$  the  $2^k$  diagonal block matrices  $M_k^j$  require  $O(\frac{1}{4} \times 2^k n^3)$  scalar multiplications for their inverses by applying LU method to each block of  $M_k^j$ . Furthermore, the  $2^k$  matrix products of the form  $X^{-1}Y$  or  $YX^{-1}$ , derived by multiplying the matrices of (3.2), require  $O(6 \cdot \frac{1}{4} \cdot 2^k n^3)$  scalar multiplications by block matrix multiplication. Now, by taking into account the property (1) and the results of Table 1 for the backward recurrence procedure we see that at Step  $i$  ( $i = 1, \dots, k - 1$ ) we need  $O(6 \cdot \frac{1}{4} (2^{k-i})^3 2^{i-1} n^3)$  scalar multiplications.

Table 1  
Numerical aspects on the recursive algorithm

Steps	Forward recurrence	Backward recurrence	
	Number of matrices to be inverted	Order of the involved matrices	Multiplications of the form $X^{-1}Y$ or $YX^{-1}$
1	$2^1$	$2^{k-1}n$	$6 \cdot 2^0$
2	$2^2$	$2^{k-2}n$	$6 \cdot 2^1$
...	...	...	...
$i$	$2^i$	$2^{k-i}n$	$6 \cdot 2^{i-1}$
...	...	...	...
$k$	$2^k$	$2^0n$	$6 \cdot 2^{k-1}$

Thus, the computational complexity  $C_R(k, n)$  of the recursive algorithm is given by

$$C_R(k, n) = O\left(7 \cdot \frac{1}{4} \cdot 2^k n^3 + \sum_{i=1}^{k-1} 6 \cdot \frac{1}{4} (2^{k-i})^3 2^{i-1} n^3\right) = O\left(\frac{1}{4} (2^{3k} + 3 \cdot 2^k) n^3\right). \quad \square$$

Then, we conclude that

**Theorem 4.2.** *Let  $A = [A_{ij}]$  be an invertible  $m \times m$  block-wise centrosymmetric matrices of order  $mn$ . Then, the computational complexity  $C(k, m, n)$  of*

the inverse of  $A$ , is given by

$$C(k, m, n) = O\left(\frac{1}{4}(2^{3k} + 3 \cdot 2^k)n^3\right).$$

*Remark 4.1.* The complexity  $C_{LU}(m, n)$  of the direct LU decomposition method for the inverse of matrix  $A$  is of order

$$C_{LU}(m, n) = O(m^3n^3). \quad (4.1)$$

Referring to the computational complexity, we see from (4.1) that the recursive inverse works best for  $m = 2^k$ . For these choices the matrix  $\Lambda$  does not need to be enlarged to the matrix  $\Delta$ . Hence, the proposed recursive inverse is about 4 times cheaper than the standard method.

## 5. Numerical experiments

We present numerical experiments for the comparison of the recursive inverse and the direct LU decomposition method with respect to accuracy and execution time.

All our computations have been done using MATLAB 7.6.0(R2008a) with unit roundoff  $u = 2^{-53} \approx 1.1 \times 10^{-16}$  and executed in an Intel Pentium M Processor 740, 1.73 GHz with 1 GB of RAM.

### Accuracy

In Table 2 we compute the relative residual inverse error

$$\frac{\|AA^{-1} - I\|_F}{\|AA^{-1}\|_F},$$

where  $I$  is the identity matrix of order  $mn$  and  $A$  is an  $mn \times mn$  block-wise centrosymmetric matrix composed of random complex elements. The inverse  $A^{-1}$  is computed by the recursive inverse and the direct LU method. The two inverse methods show a comparable accuracy. Thus, the accuracy does not actually depend on the application of a structured (recursive inverse) or a non-structured (direct LU decomposition) inverse.



Table 2  
Relative residual inverse errors of recursive and LU inverses

n	m	Relative residual error	
		Recursive inverse	LU
20	2	5.3231e-13	8.0538e-14
	4	5.9582e-12	1.7830e-11
	6	1.1033e-09	1.8875e-10
	8	5.4842e-11	1.9473e-11
40	2	6.0798e-12	8.5858e-12
	4	1.1203e-10	1.6688e-10
	6	4.8868e-10	2.0529e-10
	8	1.5980e-10	3.7244e-10
60	2	7.2340e-11	4.9961e-11
	4	1.5718e-09	3.9678e-10
	6	2.9169e-10	7.1291e-09
	8	6.5298e-09	5.3043e-09
80	2	1.1795e-10	2.0195e-09
	4	1.1437e-09	1.1071e-09
	6	3.7290e-09	1.4797e-09
	8	1.2843e-08	9.0205e-09

**Execution time**

The execution (CPU) time for the inverses with respect to  $n$  for (fixed)  $m = 4$  for the two methods is shown in Figure 1.

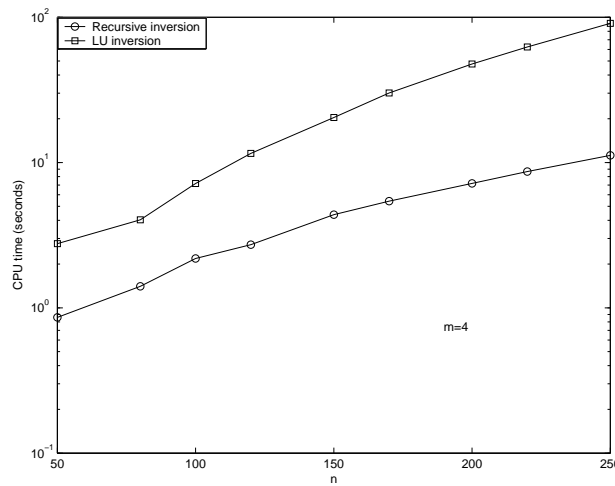


Figure 1. CPU time for the inverses with respect to  $n$  for  $m = 4$  in logarithmic scale.

The execution (CPU) time for the inverses with respect to  $m$  for (fixed)  $n = 20$  for the two methods is shown in Figure 2.

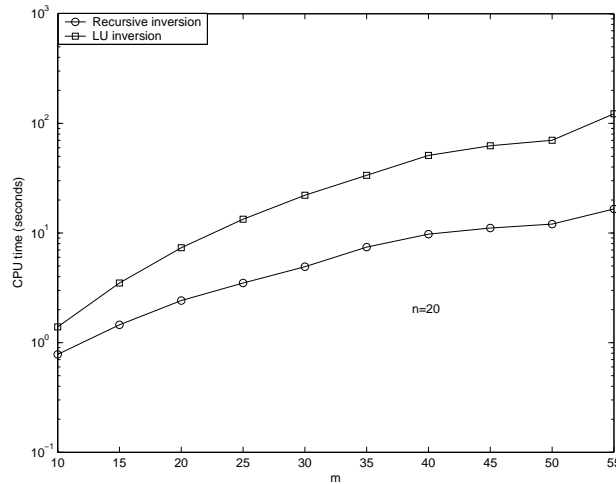


Figure 2. CPU time for the inverses with respect to  $m$  for  $n = 20$  in logarithmic scale.

It is evident by the statements of Figs. 1 and 2, the recursive inverse is clearly faster than the direct LU inverse.

ACKNOWLEDGEMENT. The authors would like to thank the editor and the referees for their helpful comments and valuable suggestions for improving this manuscript.

## References

- [1] F. A. COSTABILE and F. DELL'ACCIO, On the nonsingularity of a special class of centrosymmetric matrices arising in spectral methods in BVPs, *Appl. Math. Comput.* **206** (2008), 991–993.
- [2] J. R. WEAVER, Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues and eigenvectors, *Am. Math. Mon.* **92** (1985), 711–717.
- [3] L. DATTA and S. D. MORGERA, On the reducibility of centrosymmetric matrices, *Circ. Syst. Signal. Pr.* **8** (1989), 71–96.
- [4] Z. Y. LIU, Some properties of centrosymmetric matrices, *Appl. Math. Comput.* **141** (2003), 297–306.
- [5] I. J. GOOD, The inverse of a centrosymmetric matrix, *Technometrics* **12** (1970), 925–928.
- [6] W. C. PYE, T. L. BOULLION and T. A. ATCHISON, The pseudoinverse of a centrosymmetric matrix, *Linear Algebra Appl.* **6** (1973), 201–204.

- [7] Z. L. TIAN and C. Q. GU, The iterative methods for centrosymmetric matrices, *Appl. Math. Comput.* **18** (2007), 902–911.
- [8] Z. Y. LIU, H. D. CAO and H. J. CHEN, A note on computing matrix-vector products with generalized centrosymmetric (centrohermitian) matrices, *Appl. Math. Comput.* **169** (2005), 1332–1345.
- [9] I. S. PRESSMAN, Matrices with multiple symmetry properties: applications of centrohermitian and perhermitian matrices, *Linear Algebra Appl.* **284** (1998), 239–258.
- [10] F. L. LI, X. Y. HU and L. ZHANG, Left and right inverse eigenpairs problem of generalized centrosymmetric matrices and its optimal approximation problem, *Appl. Math. Comput.* **212** (2009), 481–487.
- [11] F. CHEN and Y. L. JIANG, On HSS and AHSS iteration methods for positive definite Toeplitz linear systems, *J. Comput. Appl. Math.* **234** (2010), 2432–2440.
- [12] A. P. LIAO, Z. Z. BAI and Y. LEI, Best approximate solution of matrix equation  $AXB + CYD = E$ , *SIAM J. Matrix Anal. Appl.* **27:3** (2005), 675–688.
- [13] A. P. LIAO and Z. Z. BAI, The constrained solutions of two matrix equations, *Acta Mathematica Sinica, English Series* **18** (2002), 671–678.
- [14] A. P. LIAO and Z. Z. BAI, Least-squares solution of  $AXB = D$  over symmetric positive semidefinite matrices  $X$ , *J. Comput. Math.* **21** (2003), 175–182.
- [15] A. P. LIAO and Z. Z. BAI, Least squares symmetric and skew-symmetric solutions of the matrix equation  $AXA^T + BYB^T = C$  with the least norm, *Mathematica Numerica Sinica* **27** (2005), 81–95 (In Chinese).
- [16] N. L. TSITSAS, E. G. ALIVIZATOS, and G. H. KALOGEROPOULOS, A recursive algorithm for the inversion of matrices with circulant blocks, *Appl. Math. Comput.* **188** (2007), 877–894.
- [17] J. BAKER, F. HIERGEIST and G. TRAPP, A recursive algorithm to invert multiblock circulant matrices, *Kyungpook Math. J.* **28** (1988), 45–50.
- [18] N. K. MADSEN, Cyclic odd-even reduction for symmetric circulant matrices, *Linear Algebra Appl.* **51** (1983), 17–35.
- [19] C. B. LU and C. Q. GU, The computation of the square roots of circulant matrices, *Appl. Math. Comput.* **217** (2011), 6819–6829.

CHENGBO LU  
 DEPARTMENT OF MATHEMATICS  
 LISHUI UNIVERSITY  
 LISHUI, 323000  
 P.R. CHINA

*E-mail:* lu.chengbo@yahoo.com.cn

CHUANQING GU  
 DEPARTMENT OF MATHEMATICS  
 SHANGHAI UNIVERSITY  
 SHANGHAI, 200444  
 P.R. CHINA

*E-mail:* cqgu@staff.shu.edu.cn

(Received September 9, 2011; revised March 6, 2012)